UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,800	10/27/2003	Hideo Miyake	1450.1005D	1082

21171 7590 02/16/2006

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
GEIB, BENJAMIN P	

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 02/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/692,800	MIYAKE ET AL.	
	Examiner	Art Unit	
	Benjamin P. Geib	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☒ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *10/27/2003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-33 have been examined.

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Application on 10/27/2003.

Priority

3. Acknowledgment is made of applicant's claim for foreign priority based on an application filed in Japan on 11/30/1999. It is noted, however, that applicant has not filed a certified copy of the 11-341077 application as required by 35 U.S.C. 119(b).

4. Acknowledgment is made of applicant's claim for foreign priority based on an application filed in Japan on 10/29/1999. It is noted, however, that applicant has not filed a certified copy of the 11-309598 application as required by 35 U.S.C. 119(b).

Double Patenting

5. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

6. Claims 1-13 are rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-13 of prior U.S. Patent No. 6,681,280. This is a double patenting rejection.

Claim Rejections - 35 USC § 112

12. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

13. Claims 15-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

14. Claim 15 recites the limitation "a condition determination section for determining whether or not a condition of the read-out conditional instruction is satisfied" in the third paragraph of the claim. There is insufficient antecedent basis for this limitation in the claim. There is no reference previously in claim 15 to a "read-out conditional instruction" only a reference to a "conditional instruction" (first paragraph of claim 15). Therefore, the phrase "a condition determination section for determining whether or not a condition of the read-out conditional instruction is satisfied" will be interpreted as "a condition determination section for determining whether or not a condition of the conditional instruction is satisfied" for the remainder of the examination.

15. All claims rejected by 35 U.S.C. 112, second paragraph, that have not been specifically addressed above are rejected on the basis of dependence.

Claim Rejections - 35 USC § 102

16. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

17. Claims 14, 15-20, and 33 are rejected under 35 U.S.C. 102(b) as being anticipated by Krauskopf, U.S. Patent No. 5,165,027.

18. Referring to claim 14, Krauskopf has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction *(breakpoint instruction)*, said apparatus comprising:

a break detection section *(32-bit register and comparator; Fig. 2, component 34)* for detecting a breakpoint set at an arbitrary position of an instruction sequence *[The comparator detects a breakpoint by comparing the address (Fig. 2; component 19) with the addresses stored in the 32-bit debug register; See column 3, lines 43-63]*;

a condition determination section *(enable logic circuit and control register; Fig. 2, components 36 and 32)* for determining whether or not a condition of said conditional instruction is satisfied *[The enable logic circuit determines whether the breakpoint is enabled (i.e. a condition of the breakpoint is satisfied) using the information store in the control register; See column 4, lines 15-39]*; and

a control section *(AND gate; Fig. 2, component 40)* for controlling a break-interrupt on the basis of a breakpoint detection result from said break detection section and a determination result from said condition determination section *[The AND gate ANDs the hit signal and enable clock signal to control the break-interrupt (i.e. breakpoint/interrupt signal; See Fig. 2) column 5, lines 1-8].*

19. Referring to claim 15, Krauskopf has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction *(breakpoint instruction)*, said apparatus comprising:

an instruction break detection section *(32-bit register and comparator; Fig. 2, component 34)* for detecting an instruction break in accordance with whether or not an instruction corresponding to an instruction address representing a breakpoint, which is set in a register, is read out, and outputting a detection signal *(hit signal; Fig. 2, component 46)* representing a detection result *[The comparator detects a breakpoint by comparing the address (Fig. 2; component 19) with the addresses stored in the 32-bit debug register; See column 3, lines 43-63]*;

a condition determination section *(enable logic circuit and control register; Fig. 2, components 36 and 32)* for determining whether or not a condition of the conditional instruction is satisfied, and outputting a determination signal *(enable clock; Fig. 2, component 44)* representing a determination result *[The enable logic circuit determines whether the breakpoint is enabled (i.e. a condition of the breakpoint is satisfied) using the information store in the control register; See column 4, lines 15-39]*; and

a logical operation section *(AND gate; Fig. 2, component 40)* for performing AND operation to said detection signal output *(hit signal; Fig. 2, component 46)* from said instruction break detection section and said determination signal output *(enable clock; Fig. 2, component 44)* from said condition determination section, and sending a break-interrupt notification *(breakpoint/interrupt signal; See Fig. 2)* in accordance with the AND operation result *[The AND gate ANDs the hit signal and enable clock signal to control the break-interrupt (i.e. breakpoint/interrupt signal; See Fig. 2) column 5, lines 1-8].*

20. Referring to claim 16, Krauskopf has taught an apparatus according to claim 15, wherein

said condition determination section *(enable logic circuit and control register)* is designed to determine whether or not an instruction word is said conditional instruction *(The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34)*, if said instruction word is said conditional instruction, determine whether or not the condition is satisfied, and output the determination signal representing the result *[The enable logic circuit determines whether the breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39]*, and

when an instruction word corresponding to the instruction address representing said breakpoint is an unconditional instruction or a conditional instruction having an unsatisfied condition, said logical operation section *(AND gate)* does not send a break-interrupt notification, and when said instruction word is the conditional instruction having

a satisfied condition, said logical operation section sends said break-interrupt notification *[When the breakpoint has an unsatisfied condition the enable clock signal will not be set (column 4, lines 30-39) and, therefore, the AND gate will not send a break-interrupt notification. The Examiner notes that the claim language necessitates only that the logical operation section does not send a break-interrupt notification when either the breakpoint is unconditional or conditional having an unsatisfied condition. When the breakpoint has a satisfied condition the enable clock signal will be set (column 4, lines 30-39). Since the hit and match signals will also be set, the AND gate will send a break-interrupt notification].*

21. Referring to claim 17, Krauskopf has taught an apparatus according to claim 15, wherein

said apparatus further comprises a mode setting section *(control register; Fig. 2, component 32)* for setting one of a first mode *(breakpoint address represents a program reference)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of said conditional instruction is satisfied, and a second mode *(breakpoint address represents a data reference)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[In both first and second modes a break-interrupt is generated when both the condition is satisfied (i.e. the clock enable is set) and the generation condition (i.e. the hit and match signals are set)]*,

said condition determination section *(enable logic circuit and control register)* is designed to determine whether or not an instruction word is said conditional instruction

(The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34), if said instruction word is said conditional instruction, determine whether or not the condition is satisfied, and output the determination signal representing the result *[The enable logic circuit determines whether the breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39]*, and

in said first mode, when an instruction word corresponding to the instruction address representing said breakpoint is an unconditional instruction or a conditional instruction having an unsatisfied condition, said logical operation section *(AND gate)* does not sends a break-interrupt notification, and when said instruction word is the conditional instruction having a satisfied condition, said logical operation section sends a break-interrupt notification, and in said second mode, when said instruction word is an instruction word corresponding to the instruction address representing said breakpoint, said logical operation section *(AND gate)* sends said break-interrupt notification *(In both the first and second modes the operation of the logical operation section is the same; See arguments for last section of claim 16 regarding its operation)*.

22. Referring to claim 18, Krauskopf has taught an apparatus according to claim 15, wherein

said condition determination section *(enable logic circuit and control register)* is designed to determine whether or not an instruction word is said conditional instruction *(The enable logic circuit receives bus control signals which allow it to determine whether*

or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34), if said instruction word is said conditional instruction, determine whether or not the condition is satisfied, and output the determination signal representing the result *[The enable logic circuit determines whether the breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39]*, and

when an instruction word corresponding to the instruction address representing said breakpoint is a conditional instruction having an unsatisfied condition, said logical operation section does not send a break-interrupt notification *[When the breakpoint has an unsatisfied condition the enable clock signal will not be set (column 4, lines 30-39) and, therefore, the AND gate will not send a break-interrupt notification]*, and when said instruction word is an unconditional instruction or the conditional instruction having a satisfied condition, said logical operation section sends said break-interrupt notification *[When the breakpoint has a satisfied condition the enable clock signal will be set (column 4, lines 30-39). Since the hit and match signals will also be set, the AND gate will send a break-interrupt notification. The Examiner notes that the claim language necessitates only that the logical operation section sends a break-interrupt notification when either the breakpoint is unconditional or conditional having a satisfied condition].*

23. Referring to claim 19, Krauskopf has taught an apparatus according to claim 15, wherein

said apparatus further comprises a mode setting section *(control register; Fig. 2, component 32)* for setting one of a first mode *(breakpoint address represents a program*

reference) in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of said conditional instruction is satisfied, and a second mode *(breakpoint address represents a data reference)* in which said break-interrupt is generating when said generation condition of said instruction break is satisfied *[In both first and second modes a break-interrupt is generated when both the condition is satisfied (i.e. the clock enable is set) and the generation condition (i.e. the hit and match signals are set)]*,

said condition determination section *(enable logic circuit and control register)* is designed to determine whether or not an instruction word is said conditional instruction *(The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34)*, if said instruction word is said conditional instruction, determine whether or not the condition is satisfied, and output the determination signal representing the result *[The enable logic circuit determines whether the breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39]*, and

in said first mode, when an instruction word corresponding to the instruction address representing said breakpoint is a conditional instruction having an unsatisfied condition, said logical operation section does not send a break-interrupt notification, and when said instruction word is an unconditional instruction or the conditional instruction having a satisfied condition, said logical operation section sends a break-interrupt notification, and in said second mode, when said instruction word is an instruction word

corresponding to the instruction address representing said breakpoint, said logical operation section sends said break-interrupt notification *(In both the first and second modes the operation of the logical operation section is the same; See arguments for last section of claim 18 regarding its operation)*.

24. Referring to claim 20, Krauskopf has taught an apparatus according to claim 15, wherein said data processing system comprises one of a scalar processor for performing one unit of processing in accordance with one instruction *(See column 3, lines 9-24 and Fig. 1)*, a long instruction word processor for parallelly executing short instructions forming a long instruction word, and a parallel processor for parallelly executing at least one basic instruction forming a variable-length instruction word.

25. Referring to claim 33, Krauskopf has taught an interrupt control method for controlling a break-interrupt in a data processing system having a function of executing a conditional instruction *(breakpoint instruction)*, said method comprising the steps of:

detecting a breakpoint set at an arbitrary position of an instruction sequence *[The comparator (Fig. 2, component 34) detects a breakpoint by comparing the address (Fig. 2; component 19) with the addresses stored in the 32-bit debug register (Fig. 2, component 34); See column 3, lines 43-63]*;

determining whether or not a condition of said conditional instruction is satisfied *[The enable logic circuit determines whether the breakpoint is enabled (i.e. a condition of the breakpoint is satisfied) using the information store in the control register; See column 4, lines 15-39]*; and

controlling the break-interrupt on the basis of a detection result of said breakpoint and a determination result of said conditional instruction *[The AND gate ANDs the hit signal and enable clock signal to control the break-interrupt (i.e. breakpoint/interrupt signal; See Fig. 2) column 5, lines 1-8].*

26. Claims 21-32 are rejected under 35 U.S.C. 102(e) as being anticipated by Alverson et al., U.S. Patent No. 6,480,818 (Herein referred to as Alverson).

27. Referring to claim 21, Alverson has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction, said apparatus comprising:

an instruction break detection section *(target thread execution subroutine; Fig. 11, component 1100)* for detecting an instruction break in accordance with whether or not an instruction corresponding to an instruction address representing a breakpoint, which is set in a register, is read out *[The break instruction, which corresponds to an address and is set in a register (See Fig. 4B), is inherently read out for instruction execution]*, and sending a break-interrupt notification in accordance with a detection result *[The target thread execution subroutine detects an instruction break (i.e. breakpoint; See Fig. 11, component 1110) and notifies the breakpoint handler subroutine (i.e. sends a break-interrupt notification); See column 21, lines 15-39 and Fig. 11]*; and

a control section *(breakpoint handler subroutine; Fig. 12, component 1125)* for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said instruction break detection section *(target thread execution subroutine)*, determining whether or not a condition of said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with a determination result *[The breakpoint handler subroutine determines if a condition of the conditional instruction is satisfied (and, therefore, the breakpoint is valid) and notifies the nub (i.e. controls break-interrupt processing) if the condition is satisfied; See column 21, lines 51-66].*

28. Referring to claim 22, Alverson has taught an apparatus according to claim 21, wherein said control section determines, in said interrupt handler, whether or not an instruction word as an instruction break target is said conditional instruction *(The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether or not a condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having an unsatisfied condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is a conditional instruction having a satisfied condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12,*

component 1240), which performs break-interrupt processing; See column 22, lines 45-65].

29. Referring to claim 23, Alverson has taught an apparatus according to claim 21, wherein

said apparatus further comprises a mode setting section *(nub thread execution routine; Fig. 5, component 500)* for setting one of a first mode *(mode when breakpoint set is a conditional breakpoint)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of said conditional instruction is satisfied, and a second mode *(mode when breakpoint set is an unconditional breakpoint)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[The nub thread execution routine sets a breakpoint mode by recording information indicating whether the inserted breakpoint is conditional or not; column 15, lines 43-63]*, and

in said first mode, said control section determines, in said interrupt handler, whether or not an instruction word as an instruction break target is said conditional instruction *(See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether or not a condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having an unsatisfied condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is a conditional instruction having a

satisfied condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*, and

in said second mode, said control section performs said break-interrupt processing when receiving said break-interrupt notification *[In the second mode the breakpoint set is unconditional and, therefore, the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*.

30. Referring to claim 24, Alverson has taught an apparatus according to claim 21, wherein said control section determines, in said interrupt handler, whether or not an instruction word as an instruction break target is said conditional instruction *(The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether a condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is a conditional instruction having an unsatisfied condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having a satisfied condition, performs said break-interrupt processing *[When the breakpoint is unconditional or conditional and the condition is satisfied (i.e. true) the breakpoint handler subroutine*

notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].

31. Referring to claim 25, Alverson has taught an apparatus according to claim 21, wherein

said apparatus further comprises a mode setting section *(nub thread execution routine; Fig. 5, component 500)* for setting one of a first mode *(mode when breakpoint set is a conditional breakpoint)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of said conditional instruction is satisfied, and a second mode *(mode when breakpoint set is an unconditional breakpoint)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[The nub thread execution routine sets a breakpoint mode by recording information indicating whether the inserted breakpoint is conditional or not; column 15, lines 43-63]*, and

in said first mode, said control section determines, in said interrupt handler, whether or not an instruction word as an instruction break target is said conditional instruction *(See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether or not a condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is a conditional instruction having an unsatisfied condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having

a satisfied condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) or breakpoint is uncoditional the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*, and

in said second mode, said control section performs said break-interrupt processing when receiving said break-interrupt notification *[In the second mode the breakpoint set is unconditional and, therefore, the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*.

32. Referring to claim 26, Alverson has taught an apparatus according to claim 21, wherein said data processing system comprises one of a scalar processor for performing one unit of processing in accordance with one instruction *(See Fig. 3, component 101 and column 1, lines 20-40)*, a long instruction word processor for parallelly executing short instructions forming a long instruction word, and a parallel processor for parallelly executing at least one basic instruction forming a variable-length instruction word.

33. Referring to claim 27, Alverson has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction, said apparatus comprising:

a software break detection section *(target thread execution subroutine; Fig. 11, component 1100)* for detecting a software break in accordance with whether or not a breakpoint instruction replaced at an arbitrary position of an instruction sequence is

executed *(See Fig. 4)*, and sending a break-interrupt notification in accordance with a detection result *[The target thread execution subroutine detects a software break (i.e. breakpoint; See Fig. 11, component 1110) and notifies the breakpoint handler subroutine (i.e. sends a break-interrupt notification); See column 21, lines 15-39 and Fig. 11]*; and

a control section *(breakpoint handler subroutine; Fig. 12, component 1125)* for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said software break detection section *(target thread execution subroutine)*, determining whether or not a condition of said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with a determination result *[The breakpoint handler subroutine determines if a condition of the conditional instruction is satisfied (and, therefore, the breakpoint is valid) and notifies the nub (i.e. controls break-interrupt processing) if the condition is satisfied; See column 21, lines 51-66]*.

34. Referring to claim 28, Alverson has taught an apparatus according to claim 27, wherein said control section determines, in said interrupt handler, whether or not an instruction word as a software break target is said conditional instruction *(The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether or not a condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said software break target is an unconditional instruction or a conditional instruction

having an unsatisfied condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said software break target is a conditional instruction having a satisfied condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*.

Referring to claim 29, given the similarities between claim 23 and claim 29 the arguments as stated for the rejection of claim 23 also apply to claim 29.

35. Referring to claim 30, given the similarities between claim 24 and claim 30 the arguments as stated for the rejection of claim 24 also apply to claim 30.

Referring to claim 31, given the similarities between claim 25 and claim 31 the arguments as stated for the rejection of claim 25 also apply to claim 31.

36. Referring to claim 32, given the similarities between claim 26 and claim 32 the arguments as stated for the rejection of claim 26 also apply to claim 32.

Conclusion

37. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the

objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin P. Geib whose telephone number is (571) 272-8628. The examiner can normally be reached on Mon-Fri 8:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Huynh can be reached on (571) 272-4147. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Benjamin P Geib
Examiner
Art Unit 2181

HENRY W. H. TSAI
PRIMARY EXAMINER

2/14/06